

BANDIT Write-up (L21 -L30)

Bandit is a character in a fictional hacking scenario on a website called OverTheWire.org.

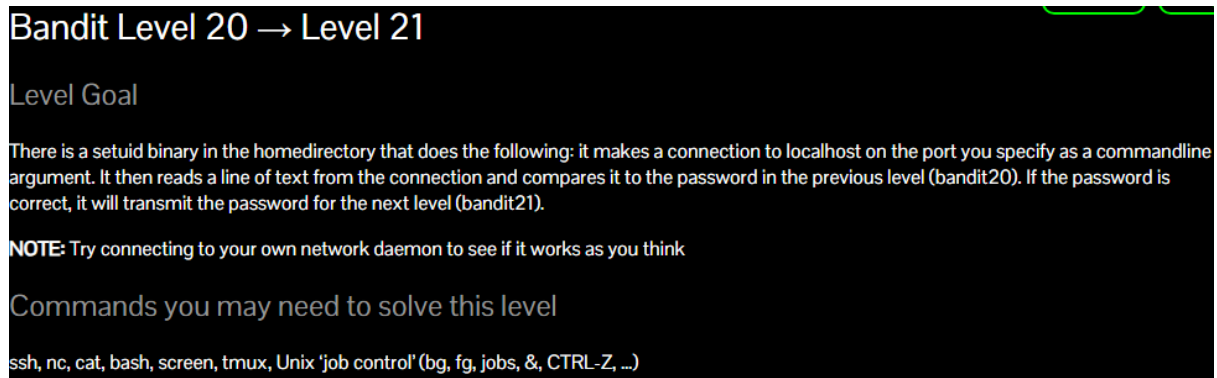
OverTheWire offers a series of wargames designed to teach cybersecurity skills in a safe environment. Bandit is the first wargame in the series, aimed at beginners. It introduces basic *Linux* commands and file manipulation through a series of challenges.

In the Bandit scenario, you play as a new user trying to gain access to higher levels by solving puzzles and cracking passwords. There is no violence or criminal activity involved.

Bandit is the suggested introductory "wargame" within the OverTheWire suite of games. It covers fundamental Linux commands and gradually progresses to advanced techniques as players advance through higher levels. Below are the walkthroughs and methodologies employed to navigate the challenges successfully.

Note for VMs: You may fail to connect to overthewire.org via SSH with a "*broken pipe error*" when the network adapter for the VM is configured to use NAT mode. Adding the setting IPQoS throughput to `/etc/ssh/ssh_config` should resolve the issue. If this does not solve your issue, the only option then is to change the adapter to Bridged mode.

Level 21:



Bandit Level 20 → Level 21

Level Goal

There is a setuid binary in the `homedirectory` that does the following: it makes a connection to localhost on the port you specify as a commandline argument. It then reads a line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password for the next level (bandit21).

NOTE: Try connecting to your own network daemon to see if it works as you think

Commands you may need to solve this level

ssh, nc, cat, bash, screen, tmux, Unix 'job control' (bg, fg, jobs, &, CTRL-Z, ...)

So in this task, we'll have two sessions at a time.

```
bandit20@bandit:~$ cat /etc/bandit_pass/bandit20 | ncat -l localhost -p 1234
EeoULMCra2q0dSkYj561DX7s1CpBu0Bt
```

```
bandit20@bandit:~$ ./suconnect 1234
Read: 0qXahG8Zj0VMN9Ghs7i0WsCfZyX0UbY0
Password matches, sending next password
```

On one side we'll cat the current level password and ncat it to our local IP address with a port number let's say 1234.

And on the other side we have suconnect, we just run this with the same port number, 1234. It will make a connection to localhost on the port you specify as a command line argument. It then reads a

line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password to the next level (bandit21).

EeoULMCra2q0dSkYj561DX7s1CpBuOBt

Level 22:

Bandit Level 21 → Level 22

Level Goal

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

Commands you may need to solve this level

cron, crontab, crontab(5) (use "man 5 crontab" to access this)

```
bandit21@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24  e2scrub_all  otw-tmp-dir  sysstat
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22
cat: /usr/bin/cronjob_bandit22: No such file or directory
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:/etc/cron.d$ cat /tmp/t706lds9S0RqQh9aMcz6ShpAoZKF7fgv
tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q
```

We have some cronjobs, cat cron...bandit22 revealed a path of the sh file. Let's cat the file and it leaves another file path. Let's cat that out and here is the flag:

tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q

Level 23:

Bandit Level 22 → Level 23

Level Goal

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

NOTE: Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

Commands you may need to solve this level

cron, crontab, crontab(5) (use "man 5 crontab" to access this)

```
bandit22@bandit:~$ cat /etc/cron.d/cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
bandit22@bandit:~$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
bandit22@bandit:~$ echo I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:~$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
0Zf11ioIjMVN551jX3CmStKLYqjk54Ga
bandit22@bandit:~$
```

The task is somewhat the same as the previous one. Now we just need to modify the echo command to print the md5sum with bandit23, so it will recognize us as bandit23 instead of 22. And change it back to md5sum. Cut -d ' ' - will cut everything after space as delimiter and -f 1 - will only print the first one.

0Zf11ioIjMVN551jX3CmStKLYqjk54Ga

Level 24:

Bandit Level 23 → Level 24

Level Goal

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

NOTE: This level requires you to create your own first shell-script. This is a very big step and you should be proud of yourself when you beat this level!

NOTE 2: Keep in mind that your shell script is removed once executed, so you may want to keep a copy around...

Commands you may need to solve this level

chmod, cron, crontab, crontab(5) (use "man 5 crontab" to access this)

The task is similar to the previous one but with some changes.

```
bandit23@bandit:~$ cd /etc/cron.d
bandit23@bandit:/etc/cron.d$ ls
cronjob_bandit22  cronjob_bandit23  cronjob_bandit24  e2scrub_all  otw-tmp-dir  sysstat
bandit23@bandit:/etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
```

We will move to cron.d there will be some files, we are interested in bandit 24 to cat, it will give us another path of .sh,

When we cat it:

```
bandit23@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname/foo
echo "Executing and deleting all scripts in /var/spool/$myname/foo:"
for i in * .*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        owner="$(stat --format "%U" ./.$i)"
        if [ "${owner}" = "bandit23" ]; then
            timeout -s 9 60 ./.$i
        fi
        rm -f ./.$i
    fi
done
```

It will give us a path. The thing is that it will execute this as bandit 23, but we need bandit 24.

```
bandit23@bandit:/tmp/anyway234$ cd /var/spool/bandit24/foo
```

We will use this command to first enter the path of bandit24 into anyway234.txt and then enter the output of the previous one into anyway234.sh:

```
bandit23@bandit:/var/spool/bandit24/foo$ echo "cat /etc/bandit_pass/bandit24 > /tmp/anyway234.txt" > anyway234.sh
```

Give it all permissions:

```
bandit23@bandit:/var/spool/bandit24/foo$ chmod 777 anyway234.sh
bandit23@bandit:/var/spool/bandit24/foo$ cat /tmp/anyway234.txt
gb8KRRcsshZXI0tUuR6ypOFjiZbf3G8
```

Cat that file after a little while to get a flag:

gb8KRRcsshZXI0tUuR6ypOFjiZbf3G8

Level 25:

Bandit Level 24 → Level 25

Level Goal

A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing. You do not need to create new connections each time

```
bandit24@bandit:~$ nc localhost 30002
I am the pincode checker for user bandit25. Please enter the password for user bandit24 and the secret pincode on a single line, separated by a space.
```

So, there's a service running on port 30002 with a password of 4 digits. That service helps find the bandit25 password. We need to brute force that service to authenticate.

We will perform this brute forcing through the iterating process. We will go through each number to authenticate between 1000 and 9999 - all 4-digit PINs.

```
bandit24@bandit:~$ for i in {0000..9999}; do echo "gb8KRRcsshZXI0tUuR6ypOFjiZbf3G8 $i"; done | nc localhost 30002
I am the pincode checker for user bandit25. Please enter the password for user bandit24 and the secret pincode on a single line, separated by a space.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
```

Use this command, to iterate through each PIN with the current password.

```
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Wrong! Please enter the correct current password and pincode. Try again.
Correct!
The password of user bandit25 is iCi86ttT4KSNe1armKiwbQNmb3YJP3q4
```

If it finds the correct one, it will display the flag:

iCi86ttT4KSNe1armKiwbQNmb3YJP3q4

Level 26:

Bandit Level 25 → Level 26

Level Goal

Logging in to bandit26 from bandit25 should be fairly easy... The shell for user bandit26 is not **/bin/bash**, but something else. Find out what it is, how it works and how to break out of it.

Commands you may need to solve this level

ssh, cat, more, vi, ls, id, pwd

So here there is a problem when we log in to the next level, because of this this time /bin/bash is missing, we need to change it back and find the flag.

```
bandit25@bandit:~$ ls
bandit26.sshkey
bandit25@bandit:~$ ls -la
total 40
drwxr-xr-x  2 root    root    4096 Jul 17 15:57 .
drwxr-xr-x 70 root    root    4096 Jul 17 15:58 ..
-rw-r----- 1 bandit25 bandit25   33 Jul 17 15:57 .bandit24.password
-r----- 1 bandit25 bandit25 1679 Jul 17 15:57 bandit26.sshkey
-rw-r----- 1 bandit25 bandit25  151 Jul 17 15:57 .banner
-rw-r--r-- 1 root     root      220 Mar 31 08:41 .bash_logout
-rw-r--r-- 1 root     root     3771 Mar 31 08:41 .bashrc
-rw-r----- 1 bandit25 bandit25   66 Jul 17 15:57 .flag
-rw-r----- 1 bandit25 bandit25    4 Jul 17 15:57 .pin
-rw-r--r-- 1 root     root      807 Mar 31 08:41 .profile
```

Let's see save passwords:

```
bandit25@bandit:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
bandit23:x:11023:11023:bandit level 23:/home/bandit23:/bin/bash
bandit24:x:11024:11024:bandit level 24:/home/bandit24:/bin/bash
bandit25:x:11025:11025:bandit level 25:/home/bandit25:/bin/bash
bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext
bandit27:x:11027:11027:bandit level 27:/home/bandit27:/bin/bash
bandit28:x:11028:11028:bandit level 28:/home/bandit28:/bin/bash
```

So there is the password that is saved in /showtext. Unlike others.

Let's cat it out:

```
bandit25@bandit:~$ cat /usr/bin/showtext
#!/bin/sh

export TERM=linux

exec more ~/text.txt
exit 0
```

So instead of bin/bash, it is set to /bin/sh.

We already have the Bandit 36 SSH key, let's try login with that:

```
bandit25@bandit:~$ ssh -i bandit26.sshkey bandit26@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit25/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit25/.ssh/known_hosts).

This is an OverTheWire game server.
More information on http://www.overthewire.org/wargames

!!! You are trying to log into this SSH server on port 22, which is not intended.

bandit26@localhost: Permission denied (publickey).
```

So, the connection ends even if we get the banner, might the problem be in between this process? We need to use *more* commands while scaling the terminal size down so that it works. At that point, we will set /bin/bash and enter into shell.

Shorten the terminal, and run this command:

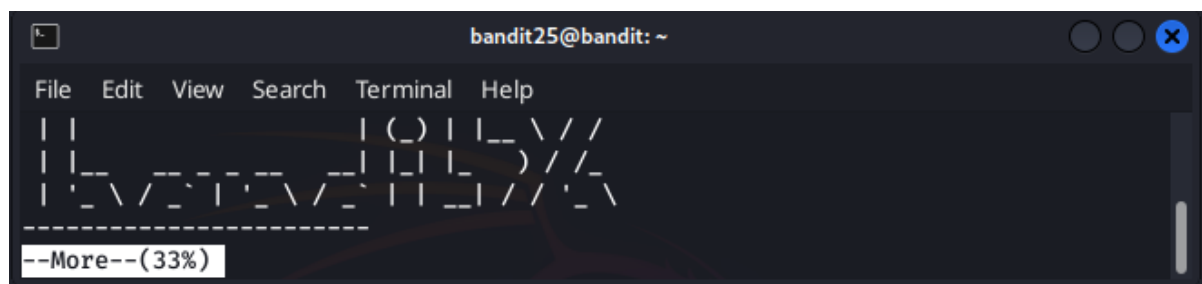

```
bandit25@bandit:~$ ssh bandit26@localhost -i bandit26.sshkey -p 2220
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit25/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit25/.ssh/known_hosts).
```

```

      _
     _/
    _/
   _/
  _/
 _/
/

```

It will have this:



Now press v to enter vim and set the default bash:

```
:set shell=/bin/bash
```

```
:shell
```

Before each command press “.”

Don't exit anything yet.

Level 27:

Bandit Level 26 → Level 27

Level Goal

Good job getting a shell! Now hurry and grab the password for bandit27!

Commands you may need to solve this level

ls

So, continue from previous level, grab the password for next level:

```
bandit26@bandit:~$ ./bandit27-do cat /etc/bandit_pass/bandit27
upsNCc7vzaRDx6oZC6GiR6ERwe1MowGB
```

We got it!

upsNCc7vzaRDx6oZC6GiR6ERwe1MowGB

Level 28:

Bandit Level 27 → Level 28

Level Goal

There is a git repository at `ssh://bandit27-git@localhost/home/bandit27-git/repo` via the port 2220. The password for the user `bandit27-git` is the same as for the user `bandit27`.

Clone the repository and find the password for the next level.

Commands you may need to solve this level

git

We just need to clone a repo, it's simple:

```
bandit27@bandit:/tmp/anyway890$ git clone ssh://bandit27-git@localhost:2220/home/bandit27-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLFXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit27/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit27/.ssh/known_hosts).
```

```
bandit27@bandit:/tmp/anyway890$ ls
repo
bandit27@bandit:/tmp/anyway890$ cd repo
bandit27@bandit:/tmp/anyway890/repo$ ls -la
total 16
drwxrwxr-x 3 bandit27 bandit27 4096 Jul 23 15:04 .
drwxrwxr-x 3 bandit27 bandit27 4096 Jul 23 15:04 ..
drwxrwxr-x 8 bandit27 bandit27 4096 Jul 23 15:04 .git
-rw-rw-r-- 1 bandit27 bandit27  68 Jul 23 15:04 README
bandit27@bandit:/tmp/anyway890/repo$ cat README
The password to the next level is: Yz9IpL0sBcCeuG7m9uQFt8ZNpS4HZRcN
```

Yz9IpL0sBcCeuG7m9uQFt8ZNpS4HZRcN

Level 29:

Bandit Level 28 → Level 29

Level Goal

There is a git repository at `ssh://bandit28-git@localhost/home/bandit28-git/repo` via the port 2220. The password for the user `bandit28-git` is the same as for the user `bandit28`.

Clone the repository and find the password for the next level.

Commands you may need to solve this level

git

```
bandit28@bandit:/tmp/anyway233$ git clone ssh://bandit28-git@localhost:2220/home/bandit28-git/repo
Cloning into 'repo'...
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Could not create directory '/home/bandit28/.ssh' (Permission denied).
Failed to add the host to the list of known hosts (/home/bandit28/.ssh/known hosts).
```

Clone the repo.

```
bandit28@bandit:/tmp/anyway233/repo$ cat README.md
# Bandit Notes
Some notes for level29 of bandit.

## credentials

- username: bandit29
- password: xxxxxxxxxx
```

There's a file in the repo, cat that out.

So there is a missing of some of the text, there may be a problem during cloning. We need to see the logs:

```
bandit28@bandit:/tmp/anyway233/repo$ git log
commit 8cbd1e08d1879415541ba19ddee3579e80e3f61a (HEAD -> master, origin/master, origin/HEAD)
Author: Morla Porla <morla@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    fix info leak

commit 73f5d0435070c8922da12177dc93f40b2285e22a
Author: Morla Porla <morla@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    add missing data

commit 5f7265568c7b503b276ec20f677b68c92b43b712
Author: Ben Dover <noone@overthewire.org>
Date:   Wed Jul 17 15:57:30 2024 +0000

    initial commit of README.md
```

It gives us a commit address in which someone added the missing files, we can use this address to to fix the missings.

```
bandit28@bandit:/tmp/anyway233/repo$ git checkout 73f5d0435070c8922da12177dc93f40b2285e22a
Note: switching to '73f5d0435070c8922da12177dc93f40b2285e22a'.
```

Now we have switched to that commit.

```
bandit28@bandit:/tmp/anyway233/repo$ cat README.md
# Bandit Notes
Some notes for level29 of bandit.

## credentials

- username: bandit29
- password: 4pT1t5DENaYuqnqvadYs1oE4QLCdjmJ7
```

Now when we cat README, it gives us the flag.

4pT1t5DENaYuqnqvadYs1oE4QLCdjmJ7

Level 30:

Bandit Level 29 → Level 30

Level Goal

There is a git repository at `ssh://bandit29-git@localhost/home/bandit29-git/repo` via the port 2220. The password for the user `bandit29-git` is the same as for the user `bandit29`.

Clone the repository and find the password for the next level.

Commands you may need to solve this level

git

Clone the repo. And cat the file:

```
bandit29@bandit:/tmp/anyway342/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: <no passwords in production!>
```

This time they didn't include the password in production.

Git works in branches. So if the password is not included in this branch, it must have been in another branch. Let's check the branches:

```
bandit29@bandit:/tmp/anyway342/repo$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/dev
remotes/origin/master
remotes/origin/sploits-dev
```

So we are in the Head branch, let's check the dev branch;

```
bandit29@bandit:/tmp/anyway342/repo$ git checkout remotes/origin/dev
Note: switching to 'remotes/origin/dev'.

You are in 'detached HEAD' state. You can look around, make experimental
```

```
bandit29@bandit:/tmp/anyway342/repo$ ls
code  README.md
bandit29@bandit:/tmp/anyway342/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: qp30ex3VLz5MDG1n91YowTv4Q8l7CDZL
```

qp30ex3VLz5MDG1n91YowTv4Q8l7CDZL